REMARKS

Entry and consideration of this Preliminary Amendment is respectfully requested.

No new matter has been added. The changes to the specification are entered only to correct translation errors and improve the English language grammar.

Respectfully submitted,

Howard L. Bernstein
Registration No. 25,665

SUGHRUE, MION, ZINN,
MACPEAK & SEAS, PLLC
2100 Pennsylvania Avenue, N.W.
Washington, D.C. 20037-3213
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

Date: May 31, 2001

APPENDIX

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION:

The specification is changed as follows:

Changes to the last paragraph of page 1 (which bridges over to page 2):

Conventionally, there are provided system simulation methods that use computers to describe elements constructing the system as linear simultaneous equations, which are repeatedly resolved to analyze operations of the system. Gaussian elimination is frequently used in the system simulation methods and is known as one of methods for obtaining solutions of the linear simultaneous equations. The Gaussian elimination produces solutions by an advance elimination process and a back-substitution process with respect to a matrix of $n \times n+1$, which is formed by linearly arranging constant terms and coefficient matrices representing coefficients for n [elements] <u>unknowns</u> of linear simultaneous equations, for example. The advance elimination process is effected to transform the matrix of $n \times n+1$ to an upper triangular matrix, while the back-substitution process is effected to sequentially produce solutions for the transformed matrix from its last row. Matrix reordering or matrix ordering is known as the technique for reordering elements (particularly, non-zero elements) of the coefficient matrices representing coefficients of the simultaneous equations in accordance with prescribed rules. Using the matrix reordering, it is possible to reduce a number of times for performing calculations such as multiplication and division that are needed for producing solutions to the linear simultaneous equations.

Changes to the second full paragraph of page 6:

FIG. 6 shows an example of a partial matrix whose elements [are not] have not been reordered yet;

Changes to the last paragraph of page 7 (which bridges over to page 8):

Using an input device (not shown), the system inputs circuit description of an electronic circuit that is a subject of simulation in step S201. The electronic circuit contains electronic elements, each of which is expressed by a linear equation. Hence, there are provided linear simultaneous equations with respect to all electronic elements contained in the electronic circuit. In step S202, positions of non-zero elements selected from among coefficients of the linear simultaneous equations are registered with a matrix A. If the electronic circuit is expressed by linear simultaneous equations of N [elements] unknowns, the matrix A is made as 'N rows×N columns'. In step S202, coefficients of the simultaneous equations are set to elements of the matrix A. In step S203, the system performs a matrix reordering process in which using matrix structure information representing values of elements of the matrix A, the system performs replacement between rows and columns in arrangement of elements of the matrix A. Herein, an order of the matrix A is determined to minimize a processing time of step S206 regarding a solving process of linear simultaneous equations. In general, a combination of elements of row i and elements of column i is called 'pivot i'. So, the step S203 proceeds to rearrangement of pivots contained in the matrix A. Namely, arrangement of elements of the matrix A is completely changed by simultaneous replacement between rows and columns in the matrix A.

Changes to the first full paragraph of page 8:

Next, the flow proceeds to step S204 in which the system produces an optimum time [split width] <u>step</u> Δt for the electronic circuit simulation. So, the system sequentially determines time points by using the time [split width] <u>step</u> Δt. For example, a next time point tn+1 is determined using Δt from a present time tn, where $tn+1=tn+\Delta t$. In step S203, the system provides matrix structure information representing new arrangement of rows and columns of the matrix A on which rearrangement of elements is completed. Based on the matrix structure information, non-linear characteristics of the electronic elements at a given time are linearized to create Newton equations (namely, linear simultaneous equations) in step S205.

Changes to the second full paragraph of page 9:

The Gaussian elimination process of step S206 can be regarded as one kind of successive elimination in which [variables] <u>unknowns</u> of simultaneous equations are successively eliminated one by one. Herein, a process for eliminating one variable is defined as a task. There exists some restriction among tasks in execution order. Such restriction can be represented by a digraph, which is called a "task graph" shown in FIG. 11. For example, there is a restriction in which task i should be executed after completion of task j. In the task graph of FIG. 11, such a restriction is translated into existence of [a side] <u>an edge</u> that directs from a vertex j to a vertex i. FIG. 11 shows a number of vertices, each of which is represented by a certain integral number encompassed by a circle. In addition, [sides] <u>edges</u> are shown by arrows, each of which represents a direction from one vertex to another vertex. Further, the processing time that is

required for completion of the task is given as a weight for the task. Such a weight is shown by an integral number that is described in proximity to the vertex corresponding to the task.

Changes to the first full paragraph of page 10:

Next, brief descriptions will be given with respect to "matrix graphs", "degrees of pivots" and "critical paths of pivots", which are required for explaining the matrix reordering process shown in FIG. 1. The matrix graph is a non-directed graph in which [a side] <u>an edge</u> exists between vertices i and j only when an element a (i,j) that lies in row i and column j of the matrix A is not zero. For example, a matrix of FIG. 6 is translated into a matrix graph of FIG. 7. Generally speaking, a term "degree" denotes a number of [sides] <u>edges</u> connected to a vertex in a graph. In the case of the matrix graph, each vertex corresponds to a pivot, hence, a degree of the vertex is called a degree of the pivot. In the matrix of FIG. 6, non-zero elements are denoted by marks × and O. Concretely speaking, numbers encompassed by circles, which are arranged on a diagonal line in the matrix, denote numbers of pivots which are originally designated prior to the matrix reordering. In addition, blank positions represent zero elements, while a symbol 'F' denotes a non-zero element that is added by the Gaussian elimination process (see step S206) with respect to each pivot.

Changes to the last paragraph of page 11 (which bridges over to page 12):

Next, the system picks up all elements (each denoted by 'e') whose values are [renewed] <u>updated</u> by the elimination process of pivot i after the aforementioned replacement. In step S310, a critical path length len(j) of pivot j to which the element e belongs are [renewed] <u>updated</u>

as len(j):=max{len(i)+w(i), len(j)}. That is, critical path lengths of pivots that are influenced by the replacement (see step S308) between pivots p and i are updated by newest values in step S310. Herein, max(x,y) is a function for selecting bigger one of variables x and y. In step S312, non-zero elements that are produced by the elimination process of pivot i are added to the matrix A. In step S314, the variable i is updated by adding '1' thereto. In step S316, a decision is made as to whether the variable i exceeds a variable N or not. Until the variable i exceeds the variable N, a series of steps S304 to S312 are repeatedly performed. Thus, the matrix reordering process is completed with respect to all pivots contained in the square matrix of N×N.

Changes to the last paragraph of page 12 (which bridges over to page 13):

Next, detailed contents of the steps S304 and S306 shown in FIG. 2 will be described with reference to FIG. 3. That is, FIG. 3 shows a flow of steps corresponding to details of the steps S304 and S306, wherein steps S402 to S412 correspond to the step S304 while steps S414 to S426 correspond to the step S306. In step S402, initialization is performed on a variable mindeg storing a minimum degree. In step S404, a variable j is set to a variable i. Then, a series of steps S406 to S412 are repeatedly performed until the variable j reaches a variable N. Specifically, in step S406, a decision is made as to whether a degree of pivot j (deg(j)) is smaller than the variable mindeg or not. If so, the flow proceeds to step S408 in which the variable mindeg is [renewed] <u>updated</u> by the degree deg(j). In step S410, the variable j is increased by '1', namely j:=j+1. In step S412, a decision is made as to whether the increased variable j is greater than the variable N or not. If not, the flow returns to step S406, so that a series of steps

S406 to S412 are to be performed again. Due to the aforementioned steps of the step S304, a minimum degree is stored in the variable mindeg.

Changes to the first full paragraph of page 13:

In the step S306 shown in FIG. 3, initialization is performed on a variable minlen that stores a minimum critical path length in step S414. In step S416, the variable j is set to the variable i. Then, a series of steps S418 to S426 are repeatedly performed until the variable j reaches the variable N. Specifically, in step S418, a decision is made as to whether the degree of pivot j (deg(j)) is smaller than mindeg+α or not. If so, the flow proceeds to step S420 in which a decision is made as to whether a critical path length of pivot j (len(j)) is smaller than the variable minlen or not. If so, the flow proceeds to step S422 in which the variable minlen is [renewed] <u>updated</u> by the critical path length len(j), and a variable p is set to the variable j. After at least one of the steps S418 to S422 is completed, the flow proceeds to step S424 in which the variable j is increased by '1', namely j:=j+1. In step S426, a decision is made as to whether the variable j is greater than the variable N or not. If not, the flow returns to step S418, so that a series of steps S418 to S426 are to be performed again. Due to the aforementioned steps of the step S306, the variable p stores a number of a pivot whose degree is under mindeg+α and whose critical path length is minimum. In addition, the variable minlen stores the minimum critical path length.